実設計に即した車載LAN FlexRay仕様解説

第3回 受信ノードにおけるデコーディング

システムの記事

佐藤道夫

今回は，送信ノードから送られてきたスロットをどのように受信ノードが受け取るのかを解説する．FlexRayでは，通信コントローラでノイズの除去やビット同期，ストローブ（ビット判定）を行う．ここでは，確実にデータを受信するためのこうしたメカニズムを説明する．（編集部）

前回（本誌2005年6月号，pp.100-107）は，フレームの中身（フォーマット）について詳しく説明しました．また，フレームを送信するための静的スロットやミニスロットの長さについて，どのようなパラメータを用いて，どう計算するのかを示しました．こうした計算は，FlexRayのネットワーク・システムを構築する際に重要となります．

今回は，送信ノードにおいてエンコーディングされたFlexRayのフレームを，受信ノードがネットワークを介してどのように受け取っているのかを見ていきたいと思います．

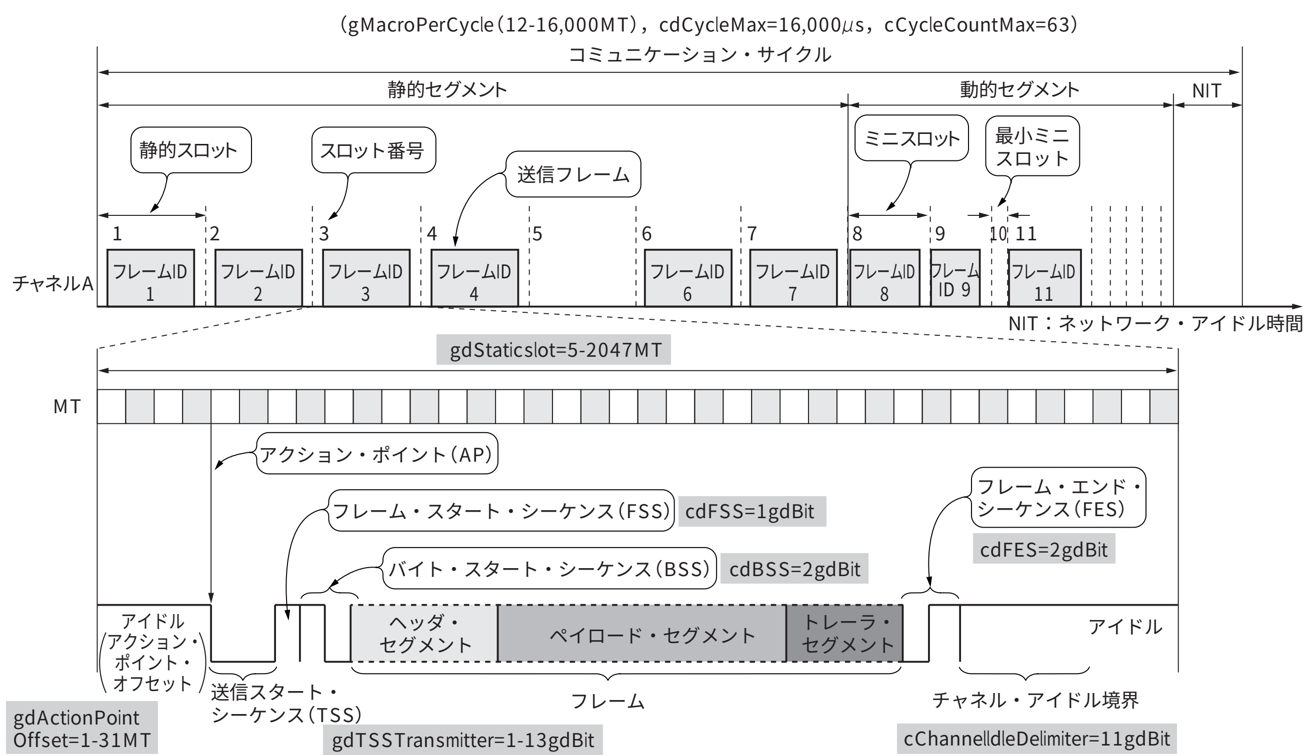


図1 スロットの送信

例えば静的セグメントのスロット１から送信されたフレームは，クラスタに接続されたすべてのノードに受信される．

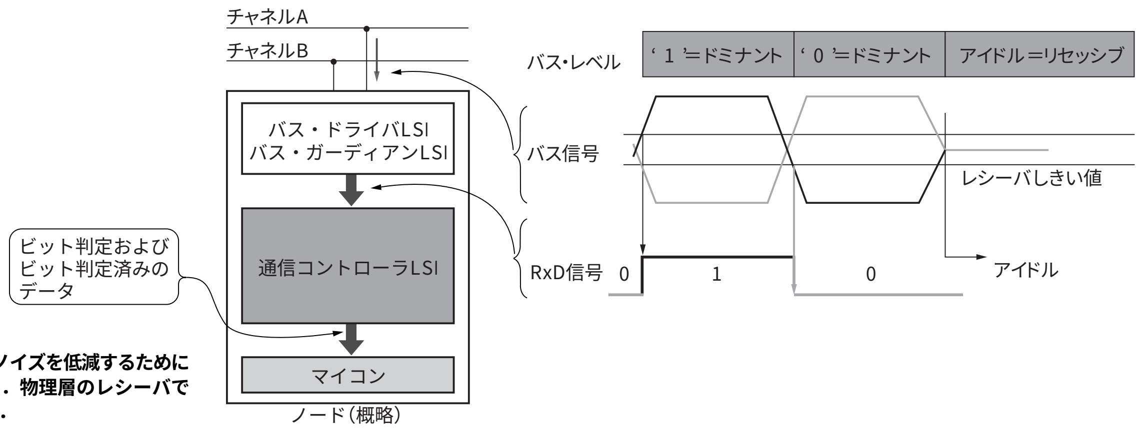


図2
データ受信
FlexRayのバス信号はノイズを低減するために差動信号となっている．物理層のレシーバでRxD信号に変換される．

1 受信ノードにおけるデータ変換

図1に示すメディア・アクセスにおいて，例えば静的セグメントのスロット1から送信されたフレームは，クラスタ（複数のノードで構成される通信システム）に接続されたすべてのノードに受信されます．すべてのノードということは，送信ノードも含まれます．

● 受信時のノイズを除去するフィルタ機能を設ける

受信ノードは，通信コントローラのRxD[注1]入力ポートに入ってきた最新の信号をいったん格納し，サンプリングします．このサンプル数を表すパラメータはcVotingSamplesであり，その値は仕様で'5'と定められています．このサンプリングのおもな目的は次のとおりです．

図2に示すように，FlexRayのバス・レベルはノイズを低減するために差動信号となっています．この信号は物理層のレシーバによって図のRxD信号に示すような波形に変換され，通信コントローラのRxD入力ポートに入ります．しかし，物理層のレシーバは波形を整形するだけなので，グリッチのような大きいノイズなどはRxD入力ポートまで届いてしまう可能性があります．そのため，通信コントローラはRxD入力ポートの信号を5サンプルずつとって，多数決で信号のレベルが'0'なのか'1'なのかを判定します（**図3**）．

なお，**図3**は，5サンプルごとに移動して信号の多数決をとっているように見えますが，実際には1サンプルごとに移動しています．つまり4サンプル前のものを使用し，最後の一つだけがつねに最新のサンプルとなっているのです．これによって，グリッチのようなノイズを除去できます．多数決判定によって整形された信号はvVotedValと呼ばれます．

● 受信ノードのビット同期をとるためのBCA

次に，BCA（bit clock alignment）という機能について説明します．BCAは受信ノードに送られてきたデータのビット同期をとるためのしくみです．

1gdBit[注2]は8サンプルで構成されています（cSamplesPerBit=8）．ネットワーク速度が10Mbpsの場合，1gdBit＝0.1μs（1÷10×10⁻⁶）なので，1サンプル時間（gdSampleClockPeriod）は0.0125μs（0.1μs÷8）になります．このとき，vVotedValのサンプリングを行うためにサンプル・カウンタが周期的に1～8（＝cSamplesPerBit）の範囲でカウ

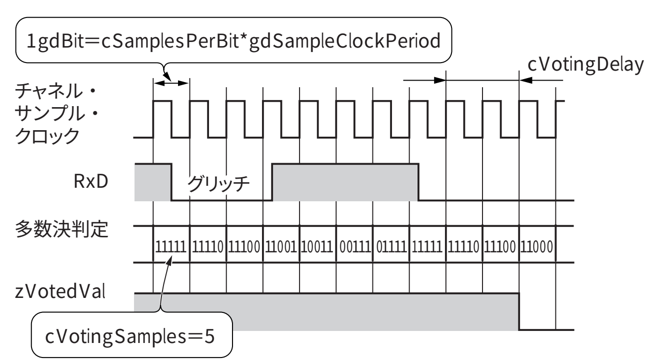


図3　サンプリングと多数決判定
各チャネル（チャネルA，チャネルB）のサンプリング・クロックごとに五つのサンプルをとる．5サンプルのうち"H"('1')のほうが多ければ，zVotedValは"H"になる．逆に"L"('0')のほうが多ければ，zVotedValは"L"になる．こうしてグリッチ（ノイズ）を除去できる．グリッチではなくRxDのレベルが変化したときはzVotedValの値も変化するが，ある遅延（cVotingDelay）を伴う．

注1：RxDはバス・ドライバからの受信データを示す．また，TxDは通信コントローラからの送信データを示す．
注2：1gdBitは，あるネットワークの伝送速度における1ビット当たりの伝送時間を表す．

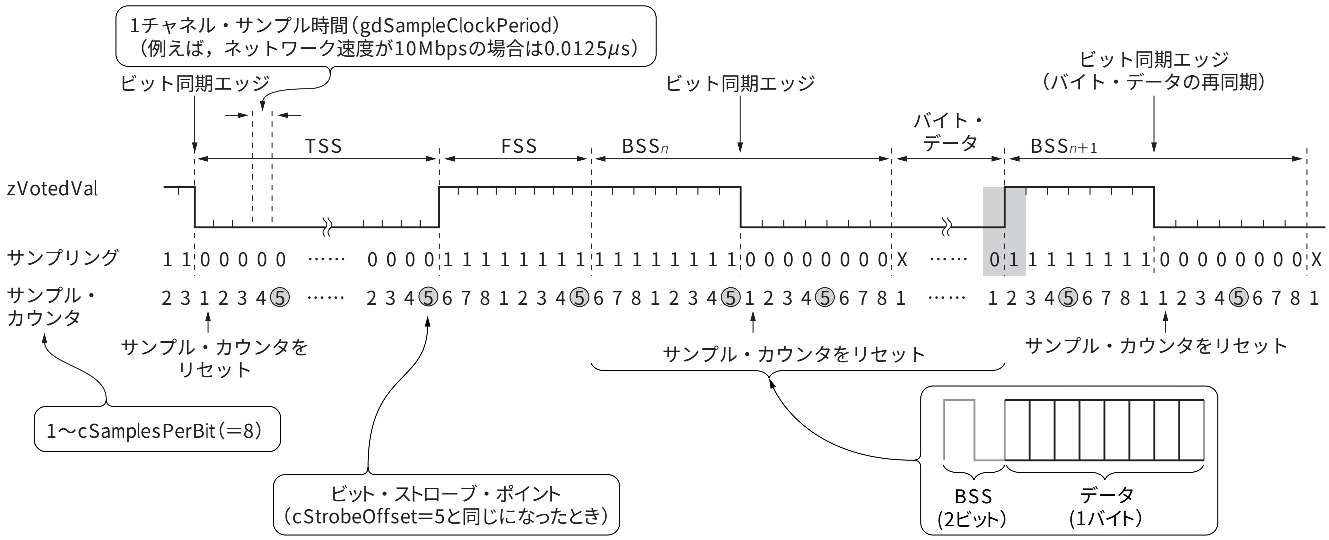


図4　ビット同期

あるスロットにおけるビット同期を示す．この例では，スロットは送信スタート・シーケンス(TSS)，フレーム・スタート・シーケンス(FSS)，バイト・スタート・シーケンス(BSS)，フレームからなる．なお，フレーム内のデータは「BSS＋1バイト」で構成される．ビット同期エッジが発生したとき，サンプル・カウンタは'1'にリセットされる．その後，サンプル・カウンタがcStrobeOffset(＝5)と同じ値になると，ビット・ストローブが行われる．

ントします．ここで受信ノードのビット・タイミングの整合性をとるために，BCAを用いてビット同期を行います．

BCAは，ローカル(各ノード)のビット・クロックに同期して行われます．また，vVotedValとして受信したビット・ストリームのストローブ信号としても使用されます．BCAは，ビット同期がイネーブル[注3]で，かつvVotedValが"H"から"L"に変化したときにのみ実行されます．つまり，vVotedValの立ち下がりエッジを検出してビット同期が行われるのです．ビット同期のエッジが検出されると，サンプル・カウンタはインクリメントされず，'1'がセットされます．

ビット同期はサンプル・カウンタだけでなく，ストローブ・ポイントの位置によって定まります．ストローブ・ポイントはサンプル・カウンタの値がcStrobeOffset(＝5)と同じ値になったときを指し，ここでビット・ストローブ(受信ビットの判定)が行われます．すなわち，サンプル・カウンタ＝cStrobeOffsetのときのvVotedValの値が，受信ビットの値として判断されます．

注3：受信ノードがエッジ検出によるビット同期をイネーブルにするのは，チャネルのアイドル状態が検出されたとき，FSS(フレーム・スタート・シーケンス)が"H"でBSS(バイト・スタート・シーケンス)の最初のビットが"H"のとき，などの条件が成立した場合である．

注4：BSSは1gdBitの"H"レベルと1gdBitの"L"レベルから構成される．BSSの立ち下がりでサンプル・カウンタは'1'にセットされるので，1バイトのデータを受信した後，サンプル・カウンタは'1'となるはずである．

図4に，ビット同期の例を示します．図の中の灰色の四角で囲んだ部分は，バイト・データを受信した後の不整合を表しています．1バイトのデータを受信した後，"H"レベルでBSSが始まりますが，このときのサンプル・カウンタの値は本来'1'となっているはずです[注4]．しかし，**図4**のケースでは，サンプル・カウンタの値は'2'となっています．つまり，ビット同期エッジが1チャネル・サンプル時間だけ遅れてデコードされています．しかし，"L"レベルのBSSに移行するとき，その立ち下がりエッジによってサンプル・カウンタはリセットされ，そこで再びビット同期がとられることになります．

● 受信の終了はチャネル・アイドル境界の検出で判定

受信ノードは，現在通信しているフレームの終了を判定するためにチャネル・アイドル境界の検出を行います．前回も説明しましたが，**図1**に示したように送信フレームの終わりを表すフレーム・エンド・シーケンス(FES)の後に11gdBit(cChannelIdelDelimiter)の"H"レベルが続きます．これがチャネル・アイドル境界です．

チャネル・アイドル境界の検出は，ノードが受信した静的スロットをデコードしている間，つねにアクティブな状態となっています(つまり，チャネル・アイドル境界の検出とビット・ストリームのデコーディングは並行して動作している)．上述したように，チャネル・アイドルである

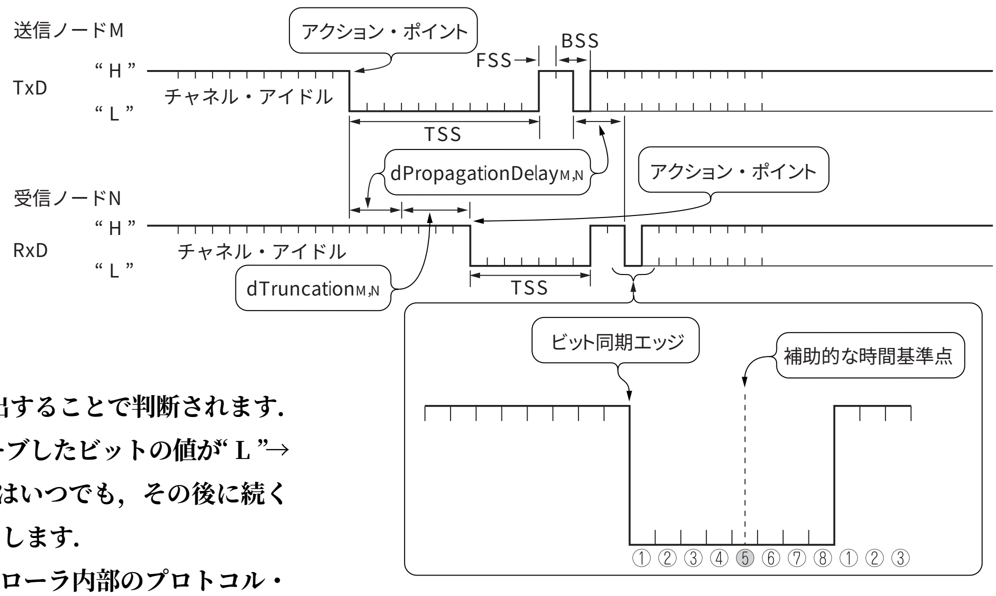


図5
TSSトランケーションと伝播遅延
クロック同期をとるためには，送信ノードと受信ノードのTSSの開始時刻の正確な時間差を測定する必要がある．TSSトランケーションなどの影響を受ける場合は，まず補助的な時間基準点（フレームの最初のBSSの2gdBit目のストローブ・ポイント）から本来の時間基準点を算出する．

かどうかは連続した“H”を検出することで判断されます．つまり，デコード中にストローブしたビットの値が“L”→“H”の順番で起こったとき注5はいつでも，その後に続く“H”となるビット数をカウントします．

かりにデコーダ（通信コントローラ内部のプロトコル・エンジン）がコーディング・エラーを検出した場合，受信ノードはチャネル・アイドル境界の検出をいったんリセットします．つまり，エラーが検出されたビットは“L”とみなし，その後，再び連続した“H”レベルのビット数をカウントします．

● 同一フレーム内における伝播遅延の違いを吸収する

前回も説明したように，アクション・ポイントとは実際にスロットの送信を開始する点のことです．

静的スロットの送受信では，送信ノードの同期フレームのアクション・ポイントと対応する受信ノードのアクション・ポイントの時間差を測定することでクロック同期をとります（クロック同期については次回に詳細を説明）．

アクション・ポイントに続いて，フレーム送信のための初期化を行うTSS（送信スタート・シーケンス）という状態に入ります．この送信側のTSSは，1～13ビット（gdTSSTransmitter）の範囲で連続した“L”レベルから構成されます．一方，受信ノードは1～14（gdTSSTransmitter＋1）ビットの範囲で連続した“L”が検出されれば，それをTSSとみなします．しかし，この受信ノードがTSSとみなしたビット数（RxD入力ポートに現れたデータ）が実際に送信されたTSSのビット数より少なくなるというケースがあります．これには，光伝送やスター・カプラ（アクティブ・スター型のネットワーク・トポロジで用いられるノードどうしを接続する装置）などの物理的な伝送媒体が影響しています．これによって，フレーム開始の最初のエッジの遅延が同じフレーム内のそのほかのエッジの遅延より長くなる可能性があります（これを「TSSトランケーション」と呼ぶ）．例えば，ノードMから送信された信号がdPropagation Delay$_{M, N}$（最大2500ns）の伝播遅延を伴ってノード*N*に受信されたとします（**図5**）．先に，クロック同期のためには送信ノードの同期フレームのアクション・ポイントと対応する受信ノードのアクション・ポイントの時間差を測定する必要があると述べましたが，TSSトランケーションが発生することで正確な時間差を知ることが不可能になります．そのため，TSSトランケーションの影響を受けない受信フレームを用いて時間計測を行わなければなりません．

受信ノードは，メッセージの最初のBSS（バイト・スタート・シーケンス）注6の間に発生する補助的な時間基準点（TRP：Time Reference Point）から時刻（zSecondary TRP）を取得し，この時刻を使って本来の時間基準点（TSSトランケーションの影響を受けていないときに受信ノードに現れるべきTSSの開始時刻）を算出します．この算出された時間基準点の時刻が，クロック同期をとるために利用されます．なお，フレームの最初のBSSの2gdBit目のストローブ・ポイントが補助的な時間基準点と定義されています．

以上のようなメカニズムによって，受信ノードは最初にグリッチなどのノイズを除去するためのサンプリングを行

注5：フレームの最後のデータ・バイトを示すフレーム・エンド・シーケンス（FES）は，1gdBitの“L”レベルと1gdBitの“H”レベルで構成される．**図1**を参照のこと．
注6：BSSは，フレーム（メッセージ）内のバイト・データの前にかならず付く．

うことで，正確なビット判定およびビット同期を行うことができます．

2 FlexRayにおけるエラーの定義

受信ノードでは，ビット判定のほかにも，フレームにエラーがあるかどうかを見ています．

FlexRayでは，エラーについて「Never give up」戦略をとっています．この戦略は，エラーによって通信を継続するか中断するかといった決定は，プロトコルではなく上位アプリケーションで行うというものです．つまり，FlexRayではCANのようにプロトコルそのものがエラーの情況に応じてバスを切り離すような仕様は盛り込まれていません．

ただしFlexRay仕様では，以下の四つのエラーが定義されています．

- シンタックス・エラー（SyntaxError）
- コンテンツ・エラー（ContentError）
- 境界違反（BViolation）
- 送信の競合（TxConflict）

次のような情況が発生するとシンタックス・エラーとなります．

- チャネルがアイドルでないのにノードが送信を開始したとき
- デコード・エラーが発生したとき
- フレームがシンボル・ウィンドウまたはネットワーク・アイドル時間でデコードされたとき
- シンボルが静的セグメント，動的セグメント，ネットワーク・アイドル時間でデコードされたとき
- 二つまたはそれ以上のフレームが一つのスロット内で受信されたとき
- 二つまたはそれ以上のシンボルがシンボル・ウィンドウ内で受信されたとき

また，コンテンツ・エラーは以下のイベントで定義されています．

- 静的セグメント上で受信したフレームのヘッダに収容されているヘッダ長がpPayloadLenghStaticと一致しないとき
- 静的セグメント上で受信したフレームのヘッダに収容されているスタートアップ・フレーム・インジケータがセットされているのに，同期フレーム・インジケータがセットされていないとき
- 静的セグメント上または動的セグメント上で受信したフレームのヘッダに収容されているフレームIDがスロット・カウンタの現在の値と一致しないとき．または動的セグメントでフレームID＝0のとき
- 静的セグメント上または動的セグメント上で受信したフレームのヘッダに収容されているサイクル・カウンタがスロット・カウンタの現在の値と一致しないとき．または動的セグメントでサイクル・カウンタ＝0のとき
- 動的セグメント上で受信したフレームのヘッダに収容されている同期フレーム・インジケータがセットされているとき
- 動的セグメント上で受信したフレームのヘッダに収容されているスタートアップ・フレーム・インジケータがセットされているとき

境界違反は，スロットの境界でチャネルがアイドルになっていない場合に発生します．また，送信の競合は，受信中にそのノードが送信を開始したときに発生します．

＊　＊　＊

ここで説明したビット同期やビットの判定は，実際のシステムではすべて通信コントローラが行います．つまり，FlexRayに対応した通信コントローラを搭載したノードであれば，受信データにノイズがのらず，同期がとれていることがハードウェア的に保証されていると言えます．ですからシステム設計者は，このことを前提としてアプリケーションの開発を行えばよいのです．

次回は，スタートアップとクロック同期について説明します．

さとう・みちお
フリースケール セミコンダクター ジャパン（株）

<筆者プロフィール>
佐藤道夫．1982年にモトローラに入社し，半導体事業部に所属．品質保証や営業を経験し，1992年からコントロール＆ネットワークのアプリケーション・エンジニアとして，LONWORKS技術の普及に従事する．現在は，フリースケール セミコンダクター ジャパンの半導体セクター トランスポーテーション＆スタンダードプロダクツグループにおいて，おもに車載向けのネットワークの技術担当マネジャーをしている．